

# **CS 595: Assignment #3**

Due on Thursday, October 2, 2014

*Dr Nelson 4:20PM*

**VICTOR NWALA**

## Contents

Problem 1	3
Problem 2	5
Problem 3	7

```

Terminal - vnwala@template: ~
File Edit View Terminal Tabs Help
42.21, 66.6.41.21
Connecting to sunshinehydrangeas.tumblr.com (sunshinehydrangeas.tumblr.com)|66.6
.42.21|:80... ^Z
[2]+  Stopped                  python downloadPage.py
vnwala@template:~$ python downloadPage.py
--2014-10-01 13:00:05-- https://www.netflix.com/entrytrap?locale=en-us
Resolving www.netflix.com (www.netflix.com)... 184.73.173.66
Connecting to www.netflix.com (www.netflix.com)|184.73.173.66|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://www.netflix.com/entrytrap?locale=en-US [following]
--2014-10-01 13:00:05-- https://www.netflix.com/entrytrap?locale=en-US
Reusing existing connection to www.netflix.com:443.
HTTP request sent, awaiting response... 200 OK
Length: 6716 (6.6K) [text/html]
Saving to: '/home/vnwala/HTML/32f9230bfef3efd44f5015725cb3a660.txt'

100%[=====] 6,716      --.-K/s   in 0s

2014-10-01 13:00:05 (955 MB/s) - '/home/vnwala/HTML/32f9230bfef3efd44f5015725cb3
a660.txt' saved [6716/6716]

--2014-10-01 13:00:05-- http://ask.fm/imsophie95
Resolving ask.fm (ask.fm)... 193.138.77.52, 193.138.77.158, 193.138.77.50, ...
Connecting to ask.fm (ask.fm)|193.138.77.52|:80... connected.

```

Figure 1: downloadPage.py at work

## Problem 1

1. Download the 1000 URIs from assignment #2. “curl”, “wget”, or “lynx” are all good candidate programs to use. We want just the raw HTML, not the images, stylesheets, etc.

Listing 1: Python script to download html pages

```

import hashlib
from hashlib import md5
import os

5 fh = open("UniqueURLS.txt", 'r')

for line in fh:
    url=line
    url=url.replace('\n', '')

10

def computeMD5hash(message):
    m = hashlib.md5()
    m.update(message)
    return m.hexdigest()

15

hashMessage = computeMD5hash(url)
print hashMessage

20

os.system("wget -O /home/vnwala/Html/" + hashMessage + ".txt " + url)

```

This script downloads html pages, saves them in text files and places them in a folder called Html. The files are also saved with their respective md5 encrypted names.

```

Terminal - vnwala@template: ~
File Edit View Terminal Tabs Help
http://dontdme.com/?reqp=1#lMap
1. http://www.godaddy.com/?ci=85889&isc=GPPT02K500

[USEMAP]
http://dontdme.com/?reqp=1#socMap
1. http://www.facebook.com/dialog/feed?app_id=115696031791232&link=https://ww
w.godaddy.com/domains/search.aspx?isc=PW999COM&picture=http://ak3.imgaft.com/ima
ges/GD_Sharehead.jpg&name=Save%20BIG%20with%20$9.99%20COMs%20from%20Go%20Daddy!
&description=Get%20your%20own%20corner%20of%20the%20Web%20for%20less!%20Register
%20a%20new%20COM%20for%20just%20$9.99%20for%20the%20first%20year%20and%20get%20
everything%20you%20need%20to%20make%20your%20mark%20online%20-%20website%20build
er,%20hosting,%20email,%20and%20more.&redirect_uri=https://www.godaddy.com/doma
ins/search.aspx?isc=PW999COM
2. https://twitter.com/intent/tweet?text=Just%20got%20a%20sweet%20deal%20from
%20%40GoDaddy.%20Picked%20up%20a%20new%20.COM%20for%20just%20%249.99%20for%20the
%20first%20year.%20Get%20yours%20while%20you%20can.&related=GoDaddy&url=http://x
.co/2qXjL
3. https://plus.google.com/share?url=http://godaddy.com/?isc=PW999COM
f9092097ee984362e02471397cb9801e
454fd0109aeccc191574273a4a04fe9c
6e18375f2789df7e06d0b3d82a8eb801
^Z
[3]+  Stopped                  python downloadText.py
vnwala@template:~$

```

Figure 2: downloadText.py at work

Listing 2: Python script to get text from html pages

```

import hashlib
from hashlib import md5
import os

5 fh = open("UniqueURLS.txt", 'r')

for line in fh:
    url=line
    url=url.replace('\n', '')

10

def computeMD5hash(message):
    m = hashlib.md5()
    m.update(message)
    return m.hexdigest()

15

hashMessage = computeMD5hash(url)
print hashMessage

20

os.system("lynx -dump -force_html " + url+ " > /home/vnwala/TextFiles/" +
    hashMessage + ".processed" + ".txt ")

```

This script downloads the text content of all the urls and stores the with the md5 names into a file.

Listing 3: Python script store URI and their md5 names

```

import hashlib
from hashlib import md5
import os

```

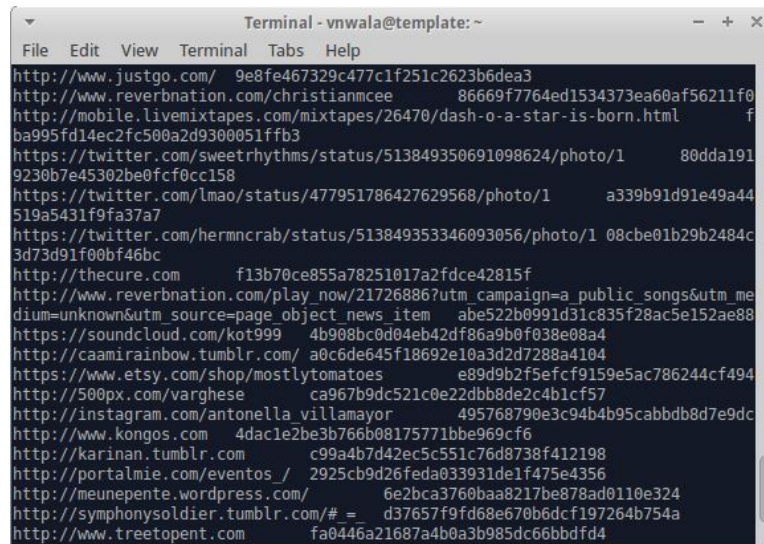


Figure 3: getHashes.py at work

```

5 fh = open("UniqueURLS.txt", 'r')

for line in fh:
    url=line
    url=url.replace('\n', '')

10

def computeMD5hash(message):
    m = hashlib.md5()
    m.update(message)
    return m.hexdigest()

15

hashMessage = computeMD5hash(url)

i = url + "\t" + hashMessage
    print i
    saveFile = open("hashes.txt", 'a')
    saveFile.write(i)
    saveFile.write('\n')
    saveFile.close()
25

```

This script converts URIs to md5 encryption and stores the mapping.

## Problem 2

2. Choose a query term (e.g., “shadow”) that is not a stop word (see week 4 slides) and not HTML markup from step 1 (e.g., “http”) that matches at least 10 documents (hint: use “grep” on the processed files). If the term is present in more than 10 documents, choose any 10 from your list. (If you do not end up with a list of 10 URIs, you’ve done something wrong).

```

Terminal - vnwala@template: ~
File Edit View Terminal Tabs Help
0.00014585764294 /home/vnwala/TextFiles/2c52d7a77747138e337e3d2435d403ba.
processed.txt
0.016967126193 /home/vnwala/TextFiles/50b726e0d5aeb66688f73b10117be629.processe
d.txt
0.00256574727389 /home/vnwala/TextFiles/09147b6690ec0c5b0b83eba13e4b27b7.
processed.txt
0.000461680517082 /home/vnwala/TextFiles/98e08de95b2f838ce86ad8fe6d531149.
processed.txt
0.00276625172891 /home/vnwala/TextFiles/5f27de776cd42edccac3ed211e9f0e25.
processed.txt
0.0505050505051 /home/vnwala/TextFiles/ef3233aefb731961481a983814b8f282.processe
d.txt
0.0013698630137 /home/vnwala/TextFiles/5aeb3f836d2dbff5f2d089e579c20179.processe
d.txt
0.00852272727273 /home/vnwala/TextFiles/60ef33641f9e8ce4acb40d4945e5b98e.
processed.txt
0.00818330605565 /home/vnwala/TextFiles/d99ad83b5dd731fa2b935f54dd8b3ad7.
processed.txt
0.00119474313023 /home/vnwala/TextFiles/79bab0513d7b21db234b6a64ba9693d1.
processed.txt
0.0011135857461 /home/vnwala/TextFiles/b7ece360eea5af79287ca1bda7f8a42f.processe
d.txt
0.00121533532206 /home/vnwala/TextFiles/1c1ca94e56bdaafc57beefca5108bcc3.
processed.txt

```

Figure 4: wordCount.py at work

Listing 4: Python script to calculate TF (wordCount.py)

```

import os
import glob
import subprocess
import re
5 from decimal import *

fh = glob.glob("/home/vnwala/TextFiles/*.txt")
for line in fh:
    url=line
    10 url=url.replace('\n','')
    proc = subprocess.Popen(["wc -w " + url], stdout=subprocess.PIPE, shell=True)
    (out, err) = proc.communicate()
    index = out.find(" ")
    index = out[:index]
    15 proc = subprocess.Popen(["grep -c 'football' " + url], stdout=subprocess.PIPE,
        shell=True)
    (out, err) = proc.communicate()
    number = out
    number=number.replace('\n','')
    if (number > "0" and index > "0"):
    20
        TF =(float(number)/int(index))
        k = str(TF) + '\t' + url
        print k
        saveFile = open('tpCal.txt','a')
    25 saveFile.write(k)
        saveFile.write('\n')
        saveFile.close()

```

This script get the total word count of my text file, queries the word “football” in all the text files, it then calculates TF for each file by diving the frequency of the word (football) by the total word count for each file and stores the result.

To calculate IDF, I used searched for the word “football”, I got 381,000,000 results. Still using the assumption that 20 billion pages are indexed by Bing:

IDF = logarithm to the base of 2 of the result of  $(20,000,000,000/381,000,000)$

IDF = 5.71407

I chose 10 URI randomly and ranked them. The table displays ranking from the highest to the lowest.

Table 1: URL RANKING

TF-IDF	TF	IDF	URI
0.078060999	0.0136612021858	5.71406519205585	http://npcironmen.com
0.046759947	0.00818330605565	5.71406519205585	http://www.maxpreps.com/national/national.htm
0.044381088	0.00776699029126	5.71406519205585	http://qoly.jp
0.028244428	0.00494296577947	5.71406519205585	http://www.chelseafc.com
0.006363106	0.0011135857461	5.71406519205585	http://www.eonline.com
0.005771783	0.0010101010101	5.71406519205585	http://twinsdaily.com/
0.0029198084	0.00051098620337	5.71406519205585	http://www.cracked.com
0.002835764	0.00049627791563	5.71406519205585	http://www.latimes.com/entertainment/
0.0026380726	0.00046168051708	5.71406519205585	http://vkfiz.ru
0.0008334401	0.00014585764294	5.71406519205585	http://www.lazywrita.com

### Problem 3

3. Now rank the same 10 URIs from question #2, but this time by their PageRank. Use any of the free PR estimators on the web, such as:

[http://www.prchecker.info/check\\_page\\_rank.php](http://www.prchecker.info/check_page_rank.php)

<http://www.seocentro.com/tools/search-engines/pagerank.html>

<http://www.checkpagerank.net/>

To answer this question, I used [http://www.prchecker.info/check\\_page\\_rank.php](http://www.prchecker.info/check_page_rank.php), to check the ranks of the respective URIs. The table displays ranking from the highest to the lowest.

Table 2: URL RANKING USING [http://www.prchecker.info/check\\_page\\_rank.php](http://www.prchecker.info/check_page_rank.php)

URI	PAGE RANK	NORMALIZED VALUES
<a href="http://www.maxpreps.com/national/national.htm">http://www.maxpreps.com/national/national.htm</a>	8/10	1
<a href="http://www.chelseafc.com/">http://www.chelseafc.com/</a>	7/10	0.875
<a href="http://www.latimes.com/entertainment/">http://www.latimes.com/entertainment/</a>	7/10	0.875
<a href="http://www.eonline.com">http://www.eonline.com</a>	7/10	0.875
<a href="http://www.cracked.com">http://www.cracked.com</a>	6/10	0.75
<a href="http://qoly.jp">http://qoly.jp</a>	5/10	0.675
<a href="http://twinsdaily.com/">http://twinsdaily.com/</a>	4/10	0.5
<a href="http://vkfiz.ru">http://vkfiz.ru</a>	2/10	0.25
<a href="http://npcironmen.com">http://npcironmen.com</a>	0/10	0
<a href="http://www.lazywrita.com">http://www.lazywrita.com</a>	0/10	0

Comparing both ranking schemes in 2 and 3, of the URIs I noticed they were not the same but in some way consistent except few exceptions. Some links rank high on both methods, some intermediate on both and others low, even if their ranking positions differ. I observed that links which have not been indexed by google

do not have a page rank using the PageRank scheme. Also the ranking system of this scheme is based on counting the number and quality of links to a page, hence giving them a rank in their order of importance.



### Check PAGE RANK of Web site pages Instantly

In order to check pagerank of a single web site, web page or domain name, please submit the URL of that web site, web page or domain name to the form below and click "Check PR" button.

Web Page URL: <http://twinsdaily.com/>

The Page Rank:  4/10

(the page rank value is 4 from 10 possible points)

Figure 5: PAGE RANK SCREEN SHOT